CHAPTER 9

PCI

This chapter presents the PC 99 requirements and recommendations for Peripheral Component Interconnect (PCI) host controllers and peripherals.

The PCI architecture has become the most common method used to extend PCs for add-on adapters. Windows and Windows NT use the basic PCI infrastructure to gain information about devices attached to the PCI bus. The ability of PCI to supply such information makes it an integral part of the Plug and Play architecture in Windows.

CardBus and device class-specific requirements related to PCI are defined in the following chapters:

- Chapter 12, “PC Card”
- Chapter 14, “Graphics Adapters”
- Chapter 15, “Video and Broadcast Components”
- Chapter 17, “Audio Components”
- Chapter 19, “Modems”
- Chapter 20, “Network Communications”

Contents

Published: June 15, 1998. Printed: 06/24/98 1:05 PM

PCI Basic Requirements

This section summarizes the basic design requirements for PCI.

9.1. All components comply with PCI 2.1

Required

Recommended: All components comply with *PCI Local Bus Specification, Revision 2.2* (PCI 2.2).

All cards, bridges, and devices that use PCI must be designed to meet the requirements defined in *PCI Local Bus Specification, Revision 2.1* (PCI 2.1), and must also comply with all Engineering Change Notices (ECNs) for PCI 2.1 approved by July 1, 1998. PCI specifications and a list of approved ECNs are available from the PCI Special Interest Group (SIG), available at http://www.pcisig.com/.

Compliance with this requirement is demonstrated based on the compliance process of the PCI SIG.

9.2. System does not contain ghost devices

Required

A computer must not include any ghost devices, which are devices that do not decode the Type 1/Type 0 indicator. Such a device will appear on multiple PCI buses.

A PCI card should be visible through hardware configuration access at only one bus/device/function coordinate.

9.3. System uses standard method to close BAR windows on nonsubtractive decode PCI bridges

Required

PCI-to-PCI bridges must comply with the *PCI to PCI Bridge Specification, Revision 1.0.* Setting the base address register (BAR) to its maximum value and the limit register to zeros should effectively close the I/O or memory window references in that bridge BAR.

9.4. System provides 3.3V to all PCI connectors

Required

PC 99 systems are required to provide 3.3 volts with amperage as defined by PCI 2.1 to all PCI connectors. This requirement enables the development of 3.3V PCI adapters without the cost of voltage regulators.

9.5. PCI add-on devices support both 5V and 3.3V signaling

Recommended

All PCI add-on devices should be Universal Boards as defined in Section 4.1 of PCI 2.1 and PCI 2.2, or be 3.3V devices that are “5V tolerant.” These devices support both 3.3V and 5V signaling, and they will allow a smooth market transition as systems change from 5V to 3.3V signaling.

This capability will be required in future versions of this design guide.

Published: June 15, 1998. Printed: 06/24/98 1:05 PM

PCI Controller Requirements

This section summarizes PCI controller requirements.

9.6. System-board bus complies with PCI 2.1

Required

Recommended: System-board bus complies with PCI 2.2.

The system-board bus hardware must comply with PCI 2.1 and should comply with PCI 2.2. The bus design must fully implement all bus requirements on every expansion card connector.

9.7. Bus master privileges are supported for all connectors

Required

To ensure full Plug and Play functionality on a PCI bus with expansion cards, all PCI connectors on the system board must be able to allow any PCI expansion card to have bus master privileges.

9.8. Functions in a multifunction PCI device do not share writable PCI Configuration Space bits

Required

The operating system treats each function of a multifunction PCI device as an independent device. As such, there can be no sharing between functions of writable PCI Configuration Space bits, such as the Command register.

Notice that the PC Card 16-bit Interface Legacy Mode BAR—offset 44h in the Type 2 PCI header—is the only exception to this requirement. This register must be shared between the two functions, just as they must share the same compatibility registers with the Exchangeable Card Architecture (ExCA) programming model, as defined in the *PCI to PCMCIA CardBus Bridge Register Description* (Yenta specification), by Intel.

For more information about design requirements for CardBus controllers, see “PC Card Socket Controller Requirements” in Chapter 12, “PC Card.”

9.9. All PCI devices complete memory write transaction (as a target) within specified times

Required

All devices must comply with the Maximum Completion Time ECN that is approved for PCI 2.1 and that is also documented in PCI 2.2.

Complying with this requirement ensures shorter transaction latencies on PCI, allowing more robust handling of isochronous streams in the system.

Published: June 15, 1998. Printed: 06/24/98 1:05 PM

Plug and Play for PCI Controllers and Peripherals

This section summarizes the Plug and Play requirements for PCI devices.

9.10. Devices use PCI 2.1 Configuration Space for Plug and Play device ID

Required

The PCI specification describes the Configuration Space used by the system to identify and configure each device attached to the bus. The Configuration Space is a 256-byte address space for each device and contains sufficient information for the system to identify the capabilities of the device. Configuration of the device is also controlled from this address space.

Configuration Space is made up of a header region and a device-dependent region. Each Configuration Space must have a 64-byte header at offset 0. All the device registers that the device uses for initialization, configuration, and catastrophic error handling must fit in the space between byte 64 and byte 255.

All other registers that the device uses during normal operation must be located in normal I/O or memory space. Unimplemented registers or reads to reserved registers must complete normally and return zero (0). Writes to reserved registers must complete normally, and the data must be discarded.

9.11. Device IDs include Subsystem IDs

Required

The Subsystem ID and Subsystem Vendor ID fields are required to comply with the Subsystem ID ECN to PCI 2.1 or the equivalent requirement in PCI 2.2. Support for these registers requires non-zero values to be populated for both the Subsystem ID and Subsystem Vendor ID.

These fields are necessary for the correct enumeration of a device. When the Subsystem ID fields are populated correctly for the adapter, Windows can differentiate between adapters based on the same PCI chip.

The Subsystem ID also allows Windows to load system miniports for system-board devices. Thus, Subsystem IDs are also a requirement on system-board devices. The exceptions to this requirement are PCI-to-PCI bridges and core chip sets.

It is required in the PCI specification and in these guidelines that all PCI functions ensure that the Subsystem IDs are in place before the operating system accesses the function’s Configuration Space registers. This is required both at initial operating system load and after any transition of the PCI bus from B3 (unpowered) back to B0.

For add-on cards, this requirement must be done by hardware on the card itself—for example, by way of serial EEPROM—and not by an extension BIOS or device driver. This is because the code is not guaranteed to run in all relevant cases, especially system sleep or dynamic bus power state transitions in which the bus becomes unpowered. Hardware methods to support this include:

- Pin strapping at Reset.
- Loading from an attached parallel or serial ROM.

Published: June 15, 1998. Printed: 06/24/98 1:05 PM

For devices that are integrated on the system board, it is acceptable for programming of the Subsystem IDs to be done by the system BIOS at power-on self test (POST) or by using ACPI control methods (_PS0 for B3 to B0 transitions). This code is always run before the operating system accesses a function's Configuration Space registers. Subsystem IDs must not be directly writable.

An acceptable implementation for Subsystem IDs that can be controlled by software is to make a copy of the Subsystem ID field and the Subsystem Vendor ID field in PCI user-defined space. Any writes to these locations will change both the copy and the original field. Any writes to the original field are discarded. POST code or ACPI control methods access the copy fields.

9.12. Configuration Space is correctly populated

Required

The following items are specific requirements for the Configuration Space:

- **9.13.1. Populate the class code register (09h) for all devices.** Follow the base class, sub-class, and programming interface values outlined in PCI 2.1 or later.
- **9.13.2. Devices must not fill BARs with random values.** See PCI 2.1 or later for correct usage of these registers. Notice that BARs 10h, 14h, 18h, 1Ch, 20h, and 24h should return zero if they are not used, indicating that no memory or I/O space is needed.

Also, for performance reasons, it is recommended that run-time registers for PCI devices should not be placed in the Configuration Space.

Note: Windows places extra constraints on a few Configuration Space registers and has uncovered some problem usage of other registers. Microsoft provides a program to help debug the use of the Configuration Space. The Pci.exe program is available at ftp://ftp.microsoft.com/developr/drg/plug-and-play/pci/pci.exe.

9.13. Interrupt routing is supported using ACPI

Required

The system must provide interrupt routing information using a _PRT object, as defined in Section 6.2.3 of the ACPI 1.0 specification.

9.14. BIOS does not configure I/O systems to share PCI interrupts

Recommended

This applies to boot devices configured by the BIOS on systems based on Intel Architecture processors. The operating system should configure all other devices. For systems that will run the Windows operating system, OEMs should design the BIOS so that it does not configure the I/O systems in the PC to share PCI interrupts for boot devices.

An exception exists for legacy audio devices following the configuration guidelines outlined in *Implementing Legacy Audio Devices on the PCI Bus*, available at http://www.intel.com/pc-supp/platform/ac97/wp/leg_pci.htm.

Windows does not support sharing an IRQ between real-mode and protected-mode code within the I/O subsystem. An example of this is an NDIS 2.0 driver (real mode) and a SCSI miniport driver (protected mode) for two PCI devices that share the same IRQ. The problem is that the IRQ needs to be reflected to real mode for the NDIS 2.0 driver to work.

However, if the IRQ is reflected to real mode, the real-mode SCSI driver, which

Published: June 15, 1998. Printed: 06/24/98 1:05 PM

usually is not called because Windows takes over in protected mode, might touch the hardware, causing the SCSI miniport to be confused. Windows resolves this problem either by switching everything to protected mode or by falling back to real mode.

9.15. BIOS configures boot device IRQ and writes to the interrupt line register

Required

This requirement applies to boot devices configured by the BIOS on systems based on Intel Architecture processors. Windows should configure all other devices because, after an IRQ is assigned by the system BIOS, Windows cannot change the IRQ. If the BIOS assigns the IRQ and Windows needs it for another device, a sharing problem occurs.

The BIOS must configure the boot device IRQ to a PCI-based IRQ and must write the IRQ into the interrupt line register 3Ch, even if the BIOS does not enable the device. This way, the operating system can still enable the device with the known IRQ at configuration time, if possible.

See also requirement 12.23, “16-bit PC Card card driver supports sharing of level-mode interrupts.”

9.16. Systems that support hot plugging for any PCI device use ACPI-based methods

Required

Hot-plugging capabilities are not required for PCI devices. Windows 98 and Windows NT 5.0 support dynamic enumeration, installation, and removal of PCI devices only if there is a supported hardware insert/remove notification mechanism.

The appropriate notification mechanism is supported as a bus standard for CardBus bus controllers. For other solutions, such as those required for docking stations or hot-plug PCI devices, the hardware insert/remove notification mechanism must be implemented as defined in Section 5.6.3 of the ACPI 1.0 specification.

Published: June 15, 1998. Printed: 06/24/98 1:05 PM

Power Management for PCI Controllers and Peripherals

This section summarizes the specific PCI power management requirements.

9.17. All PCI components comply with PCI Bus Power Management Interface specification

Required

The PCI bus, any PCI-to-PCI bridges on the bus, and all add-on capable devices on the PCI bus must comply with *PCI Bus Power Management Interface Specification, Revision 1.1* or later. This includes correct implementation of the PCI Configuration Space registers used by power management operations, and the appropriate device state (D*x*) definitions.

For the PCI bus, any PCI-to-PCI bridges on the bus, and all add-on-capable devices on the PCI bus. ACPI is not an acceptable alternative for PC 99.

9.18. System provide support for 3.3Vaux if a PCI bus is implemented

Required

System support for delivery of 3.3Vaux to the PCI bus must be capable of powering a single PCI slot with 375 mA at 3.3V and it must also be capable of powering each of the other PCI slots on the segment with 20 mA at 3.3V whenever the PCI bus is in the B3 state.

Systems must be capable of delivering 375 mA at 3.3V to all PCI slots whenever the PCI bus is in any “bus powered” state: B0, B1, or B2.

9.19. Bus power states are correctly implemented

Required

The PCI bus must be in a bus state (B*x*) no higher than the system sleeping state (S*x*). This means that if the system enters S1, the bus must be in B1, B2, or B3. If the system enters S2, the bus must be in B2 or B3, and if the system enters S3, the bus must be in B3. Of course, in S4 and S5, the system power is removed, so the bus state is B3. A PCI bus segement must not transition to the B3 state until all downstream devices have transitioned to D3..

Control of a PCI bus segment’s power is managed using the originating bus bridge for that PCI bus segment.

- For CPU-to-PCI bridges, these controls must be implemented using ACPI or the *PCI Power Management Interface Specification, Revision 1.1* or later.
- For PCI-to-PCI bridges, these controls must be implemented in compliance with the *PCI Power Management Interface Specification, Revision 1.1*. or later.

Published: June 15, 1998. Printed: 06/24/98 1:05 PM

9.20. PCI-based modem and network adapters support wake-up

Required

PCI-based modem and network adapters must support wake-up as follows:

- Modem adapters must be capable of generating a power management event (PME# assertion) from the D3 cold device state. It is recommended that modem adapters support capture of Caller ID with hardware support for the AT+VRID, “resend caller ID,” voice modem command.
- Network adapters must support the generation of a power management event (PME# assertion) from the D3 cold device state if the physical layer technology is generally capable of operating under the voltage and current constraints of the D3 cold device state. Network adapters must also support the minimum requirements for network packet filtering/wake-up capability as defined in requirement 20.56, “Device supports wake-up events.”

PCI References

The following represents some of the references, services, and tools available to help build hardware that is optimized to work with Windows operating systems.

Advanced Configuration and Power Interface Specification, Revision 1.0
http://www.teleport.com/~acpi/

Default Device Class Power Management Reference Specification, Version 1.0, and other power management specifications
http://www.microsoft.com/hwdev/onnow.htm

“Efficient Use of PCI,” Platform Architecture Labs, Intel Corporation
http://support.intel.com/oem_developer/chipsets/pci/general/pci001.htm

Implementing Legacy Audio Devices on the PCI Bus
http://www.intel.com/pc-supp/platform/ac97/wp/leg_pci.htm

Microsoft testing tools, specifications, and information, including “IDs and Serial Numbers for Plug and Play” and other related articles
E-mail: pciinfo@microsoft.com
http://www.microsoft.com/hwtest/
http://www.microsoft.com/hwdev/busbios/
ftp://ftp.microsoft.com/developr/drg/plug-and-play/pci/pci.exe

PCI Bus Power Management Interface Specification, Revision 1.1
PCI Local Bus Specification, Revision 2.1 and later
PCI to PCI Bridge Specification, Revision1.0.
PCI SIG
Phone: (800) 433-5177
http://www.pcisig.com/

PCI to PCMCIA CardBus Bridge Register Description (Yenta specification)
http://www.pc-card.com/

Published: June 15, 1998. Printed: 06/24/98 1:05 PM

Checklist for PCI

If a recommended feature is implemented, it must meet the PC 99 requirements for that feature as defined in this document.

9.1. All components comply with PCI 2.1
Required

9.2. System does not contain ghost devices
Required

9.3. System uses standard method to close BAR windows on nonsubtractive decode PCI bridges
Required

9.4. System provides 3.3V to all PCI connectors
Required

9.5. PCI add-on devices support both 5V and 3.3V signaling
Recommended

9.6. System-board bus complies with PCI 2.1
Required

9.7. Bus master privileges are supported for all connectors
Required

9.8. Functions in a multifunction PCI device do not share writable PCI Configuration Space bits
Required

9.9. All PCI devices complete memory write transaction (as a target) within specified times
Required

9.10. Devices use PCI 2.1 Configuration Space for Plug and Play device ID
Required

9.11. Device IDs include Subsystem IDs
Required

9.12. Configuration Space is correctly populated
Required

9.13. Interrupt routing is supported using ACPI
Required

9.14. BIOS does not configure I/O systems to share PCI interrupts
Recommended

9.15. BIOS configures boot device IRQ and writes to the interrupt line register
Required

9.16. Systems that support hot plugging for any PCI device use ACPI-based methods
Required

9.17. All PCI components comply with PCI Bus Power Management Interface specification
Required

9.18. System provide support for 3.3Vaux if a PCI bus is implemented
Required

9.19. Bus power states are correctly implemented
Required

9.20. PCI-based modem and network adapters support wake-up
Required

Published: June 15, 1998. Printed: 06/24/98 1:05 PM

Published: June 15, 1998. Printed: 06/24/98 1:05 PM